

ВЦ ТГУ

V

**ВСЕСОЮЗНОЕ
СОВЕЩАНИЕ
ПОЛЬЗОВАТЕЛЕЙ
ЭВМ
ТИПА
»УРАЛ«**

ТЕЗИСЫ ДОКЛАДОВ II

ТАРТУ,
1966



ТАРТУСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Вычислительный центр

ТЕЗИСЫ ДОКЛАДОВ

У ВСЕСОЮЗНОГО СОВЕЩАНИЯ ПОЛЬЗОВАТЕЛЕЙ

ЭВМ ТИПА "УРАЛ"

Секция II

Автоматизация программирования. Трансляторы с алгоритмическими языками. Обслуживающие программы.

Тарту 1966



ПРОГРАММА ПЕЧАТИ КАРТ В ИЗОЛИНИИ НА ЭВМ „УРАЛ-4“

Алаев Г.Т.

Процесс построения карт вручную довольно трудоемок и в значительной мере субъективен. Естественно, что возникла мысль использовать для этой цели ЭВМ [1, 2].

Работа ЭВМ при построении карты складывается из двух этапов:

а) интерполирование исходной информации (заданной обычно в относительно небольшом количестве неравномерно расположенных точек),

б) вывод полученных данных на печать в форме, удобной для построения карты.

В данном сообщении вопросы интерполирования не рассматриваются. Предполагается, что значения отметок, выводимых на печать, заданы в узлах равномерной квадратной сетки. Для сохранения равенства масштабов по координатным осям число точек по горизонтам утраивается, а по вертикали удваивается (учитывая, что в АЦПУ отношение расстояния между точками горизонтального к вертикальному равно 4:6). Дополнительные точки получают путем линейной интерполяции.

Карта выводится на печать в виде совокупности изолиний, отметки которых изменяются с постоянным шагом. Точки изолинии печатаются звездочкой (*) и снабжаются числовой отметкой.

Процесс отыскания местоположения точек и соответствующих им отметок протекает следующим образом.

Берется отметка первой слева точки в строке и выделяется наибольшее число n , кратное шагу. Ищут разности между этим числом n и отметками по строке до тех пор, пока эти разности убывают (по модулю). Если наименьшая из разностей меньше половины шага, то считается, что положение точки изолинии с отметкой n найдено, и печатается звездочка и число n . После этого n изменяют на величину шага в зависимости от знака последней разности и процесс повторяется.

Время печати карты составляет для квадрата 60 x 60 в среднем 10 минут.

По программе проведены опыты построения геологических структурных карт. Результаты получились, в целом, удовлетворительные.

По-видимому, применение ЭВМ при построении карт возможно не только в геологии, но и в других областях.

Литература.

1. Польстер Л.А., Зайдель А.Р. Использование ЭВМ для решения некоторых геологических задач, 1965.
2. "Труды Главной геофизической обсерватории", вып. II4.

СИСТЕМА АВТОМАТИЗАЦИИ ПРОГРАММИРОВАНИЯ

Анисимов В.И., Лозница В.С., Козлык Л.Г.,

Тодавчин А.Ю., Вуханевич М.Г.

При решении комплексных инженерных задач, из-за ограниченной оперативной памяти машины возникает необходимость разбивать их на блоки.

Данная САП предусматривает наличие этих блоков в общей памяти и позволяет с помощью специальной программы по заданной для нее информации автоматически решать комплексные задачи, используя стандартные программы, табличные исходные данные и необходимые программные блоки, находящиеся в ИМБ, а также исходные данные, вводимые с перфоленты.

Кроме того, программа позволяет автоматически запоминать в ИМБ результаты решения задач в виде таблиц для дальнейшего использования.

ОПЫТ ИСПОЛЬЗОВАНИЯ ИНТЕРПРЕТИРУЮЩЕЙ ПРОГРАММЫ ДЛЯ ЭВМ "УРАЛ-2"

Астахов Ю.И., Перфильева А.И.

В КВЦ ЦНИИ ТМАШ в течение 1965 года освоена и используется интерпретирующая система ИС-Б, разработанная Белоноговым. Она позволила существенно облегчить труд программиста, автоматизировать использование библиотеки СП. Применение ИС позволило сократить на 15 % время, необходимое на составление и отладку программ. Нами расширена библиотека СП, в нее дополнительно включены некоторые обслуживающие и отладочные программы.

Среди них:

- 1/ различные программы печати,
- 2/ программа расчета с удвоенной точностью,
- 3/ программы упорядочения последовательности чисел,
- 4/ программа вычисления y^x /позволяющая найти корень любой степени/,
- 5/ стандартная программа отладки и др.

Общее количество СП около 50.

Необходимо договориться об унификации номеров стандартных программ, выделив определенные номера для универсальных программ.

ОБСЛУЖИВАЮЩИЕ ПРОГРАММЫ ДЛЯ ЭВМ "УРАЛ-2"

Астахов Ю.И., Сергеева Е.С.

И. Программа печати чисел с их нумерацией /для облегчения поиска нужного параметра в числовом массиве/.

II. Программа специальной печати чисел с номерами ячеек, в которых они содержатся /для облегчения отладки/.

ПРЕОБРАЗОВАНИЕ ЦИФРОВОЙ ПОКОЛОНЧАТОЙ ИНФОРМАЦИИ В СТРОЧНУЮ

Афанасьева Г.И., Коростовская Г.М.

А) Весь исходный материал – бух. документы – поступает на БЭМ "Урал-4" в виде 45 колонковых перфокарт. Информация пробивается по колонкам (I-40), в каждой колонке – одна десятичная цифра.

б) При вводе в машину п/карт информация, записанная по колонкам, разворачивается согласно принятому макету в строку.

в) Необходимый контроль разворота осуществляется при помощи обратного разворота, т.е. полученная строка приводится к первоначальному виду для сравнения с исходным материалом.

ОПЕРАЦИИ С ЧИСЛАМИ, ПРЕВЫШАЮЩИМИ РАЗРЯДНУЮ СЕТКУ
"УРАЛ-4", С СОХРАНЕНИЕМ ТОЧНОСТИ

Афанасьева Г.И., Коростовская Ю.М.

ЛТК Госбанка часть работ по бухгалтерскому учету переводит на ЭВМ "Урал-4". Выполнение на машине расчетов, связанных с бухгалтерией, требует использования специальной формы записи обрабатываемого материала. Специфика бухгалтерских расчетов и объем материала исключают возможность использования как режима с фиксированной, так и режима с плавающей запятой (с любой степенью точности).

Исходя из принятой условной записи чисел в машине, составлены специальные программы алгебраического сложения и деления чисел.

СОРТИРОВКА УЧЕТНО-СТАТИСТИЧЕСКОЙ ИНФОРМАЦИИ

Афанасьева Г.И., Коростовская Е.М.

Введенный материал перед использованием необходимо упорядочить. Для этого составлены следующие программы сортировки:

- а) Сортировка методом последовательного упорядочения пар смежных записей /в пределах НФ/.
- б) Сортировка методом перестановки смежных записей по возрастанию 5 главных разрядов и при 5 одинаковых разрядах по возрастанию 3-х вспомогательных разрядов /в пределах НФ/.
- в) Поразрядная сортировка массивов записей, превышающих объем НФ, с использованием 2-х МБ и контролем сортировки.

Запись - 6 неполных ячеек НФ, определяющая порядок следования записей - I-я полная ячейка.

ОТЛАДКА ПРОГРАММЫ ПРИ РУЧНОМ СПОСОБЕ ПРОГРАММИРОВАНИЯ

Афиногенова Е.П., Альбрехген Л.К.

В докладе рассматриваются способы отладки программ, составленных вручную.

Постановка задач на машину определяется следующими моментами:

- а) выбор метода решения задачи;
- б) составление алгоритма;
- в) составление программы;
- г) отладка программ;

каждый из которых вносит свои ошибки. Основное внимание уделяется характеру ошибок, возможных при составлении программ.

Разбираются полуавтоматические средства отладки программ.

1. Искусственное разбиение программы на блоки для проверки передач управления.

2. Программа отладки методом контрольных точек. В схему программы заложен тот же принцип, что и при ручной отладке.

Программа позволяет проверить в одном или нескольких циклах в заранее указанных местах (узлах):

1) результаты всех арифметических операций (как с фиксированной, так и с плавающей запятой) в восьмеричной или десятичной системе. К арифметическим операциям должны быть

заранее вычислены контрольные точки (восьмеричные или десятичные);

2) результаты операций "I0 - I4" с контрольными точками или без них;

3) исполнение команд условной передачи управления, передачи управления по ключу, изменения счетчика циклов (команды 25, 24) и команды "30".

Программа предназначена для отладки программ с большим количеством вычислений.

Программа занимает 713_8 рабочих ячеек, включая подпрограммы перевода $I0 \rightarrow 2$ и $2 \rightarrow I0$ с плавающей и фиксированной запятой.

Программа - анализатор

Осуществляет автоматическое декодирование программы, позволяющее установить правильность составленной программы путем сличения данного алгоритма с алгоритмом, полученным с помощью "анализатора". В работе "анализатора" различаются 2 этапа.

На первом этапе путем однократного просмотра "анализатор" разбивает отлаживаемую программу на блоки.

На втором этапе он восстанавливает формулы, заключенные в арифметико-логических блоках, и выводит на устройство печати линеаризованную блок-схему программы.

Работа по составлению "анализатора" еще не закончена. В настоящее время закончен этап принципиальной проверки метода декодирования.

ПРОГРАММА "СТОРОЖ" ДЛЯ ХРАНЕНИЯ БОЛЬШИХ МАССИВОВ ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ НА "УРАЛ-4"

Барашиков

При решении экономических задач возникла необходимость хранения большого массива исходных данных на внешних устройствах (НМБ или НМЛ).

В ВЦ ХТТЗ им. С.М. Кирова составлена программа "Сторож", состоящая из двух частей: программы переписи и программы контроля массивов.

1. Программа переписи предназначена для переписи массивов с перфокарт или НМЛ на магнитный барабан (НМБ) и магнитную ленту (НМЛ).

2. Программа контроля предназначена для контроля сохранности массивов на НМБ и НМЛ, записанных с помощью программы переписи.

При работе данной программы используются внешние накопители НМБ и НМЛ, выходной перфоратор (ПФУ), печатающее устройство (ПЧУ).

Информация для работы "Сторожа" располагается на перфокарте, за которой ставятся перфокарты с массивами.

ПРОГРАММИРУЮЩАЯ ПРОГРАММА СППО

Барлас Р.А.

СППО предназначена для автоматического программирования арифметических операторов.

При подготовке задания для СППО допускаются символы: $+$, $-$, \times , $:$, $($, $)$, \sum_{α}^{β} , а также обращения к СП вычисления различных функций ($\sqrt{\quad}$, \sin , \exp, \dots).

Подготовкой задания для СППО является перечисление в естественном порядке всех встречающихся в формуле адресов и символов. Для составления задания не требуется умение программировать.

Для знакомых с программированием возможности с ППО расширяет наличие некоторых неарифметических операций: перепись массива с модификацией, передача управления на метку.

АВТОДИСПЕТЧЕР ДЛЯ ОТЛАДКИ И СЧЕТА ПРОГРАММ НЕБОЛЬШИХ РАЗМЕРОВ

Быгдай В.П.

Настоящая система автоматического программирования преследует следующие цели:

1. Максимально упростить ручное программирование на "Урале-2", приблизив его по сложности к программированию на малых машинах типа "Промитъ".

2. Автоматизировать процесс отладки.

3. Свести к минимуму действия оператора за пультом.

Данная система выполняет следующие функции:

1. Автоматический ввод, размещение, перевод и печать исходного материала.

2. Переформирование программы в случае многовариантного счета.

3. Использование системы псевдокоманд (применяется компилирующая программа с библиотекой стандартных программ Ю.Н. Ефимова - Архт. НИИ).

4. Автоматическая обработка контрольного варианта.

5. Автоматизация отладки: программа начинает работать при нажатии кнопки "пуск" и проходит до конца без остановок; лента с программой в процессе отладки не подвергается исправлениям, так как они вносятся программным путем; разработанная

система приказов позволяет исключить вмешательство оператора в процессе отладки.

УНИВЕРСАЛЬНАЯ ПРОГРАММА СИНТАКСИЧЕСКОГО АНАЛИЗА ДЛЯ ТРАНСЛЯТОРОВ С ЯЗЫКОВ, ЗАДАНИЙ ГРАФОВЫМИ ГРАММАТИКАМИ

Борщев В.Б.

1. Рассматриваются так называемые графовые грамматики, которые могут быть использованы для задания языков программирования. Показывается, что эти грамматики эквивалентны бесконтекстным грамматикам Хомского /а, следовательно, и "языковым формам"/.

2. Обсуждается метод синтаксического анализа для языков, заданных графовыми грамматиками.

3. Рассматривается универсальная программа синтаксического анализа для машины "Урал-4". Программа может быть использована:

А. для проверки синтаксической правильности программ на любом языке программирования, заданном графовыми грамматиками;

Б. как составная часть трансляторов с этих языков. Предварительный синтаксический анализ при трансляции помогает строить транслятор более экономно и сравнительно легко менять его при изменении языка.

ОБ УСОВЕРШЕНСТВОВАНИИ И ВНЕДРЕНИИ ТРАНСЛЯТОРА
С АДРЕСНОГО ЯЗЫКА НА МАШИНУ "УРАЛ-2" ("УРАЛ-4")

Бушко-Жук М.М., Гонца М.Г., Соловьева И.А., Фесенко В.И.

Транслятор с адресного языка, о котором было доложено на 4-м Всесоюзном совещании пользователей машинами типа "Урал" в течение прошедшего года внедрялся в ряде организаций. Для одной из них он в настоящее время переделывается для машины "Урал -2М".

Сообщается об изменениях во входном языке.

О СИНТАКСИЧЕСКОМ МЕТОДЕ ПРОВЕРКИ ФОРМАЛЬНОЙ ПРАВИЛЬНОСТИ ТРАНСЛИРУЕМЫХ АЛГОРИТМОВ

Бушко-Жук М.М., Фесенко В.И.

Возможны различные методы проверки формальной правильности транслируемых алгоритмов. Существует метод попарной встречаемости символов. Дается его обобщение — так называемый матричный метод. Предлагаемый второй метод — метод сворачивания.

Пусть дан алфавит A и множество всевозможных слэв в нем — всеобщий язык \mathcal{M} . В \mathcal{M} выделяем конечное подмножество S — его синтаксис. Из элементов S строится Φ — множество подстановок (элементарных сворачиваний) $\varphi: x \rightarrow y, x, y \in S$, причем длина x больше длины y . Применимость $\varphi \in \Phi$ к $a \in \mathcal{M}$ — в смысле марковских алгоритмов. Если существует сворачивание (последовательность элементарных сворачиваний), дающее применительно к $a \in \mathcal{M}$ некоторое $b \in S$, то a назовем сворачиваемым, а алгоритм, им представляемый, — формально правильным. Предлагаются общие правила построения алгоритма сворачиваний.

РЕАЛИЗАЦИЯ СИНТАКСИЧЕСКОГО МЕТОДА ПРОВЕРКИ ФОРМАЛЬНОЙ ПРАВИЛЬНОСТИ ТРАНСЛИРУЕМЫХ АЛГОРИТМОВ НА ЭВМ "УРАЛ-2"

Гамецкая С.П., Данильченко А.Ф.

Строится множество Φ подстановок (элементарных сворачиваний) $\varphi: x \rightarrow y, x, y \in S$ (S - синтаксис языка).

Это множество подстановок разбивается на три группы:

- 1) подстановки, сворачивающие части формул,
- 2) подстановки, сворачивающие формулы,
- 3) подстановки, проверяющие наличие соответствия между началами и концами циклов.

Просмотр алгоритма, записанного на входном языке, производится на рабочем столе. При просмотре применяются последовательно указанные группы подстановок.

Алгоритм записан синтаксически правильно, если в результате применения к нему всех трех групп подстановок получаем полное сворачивание алгоритма.

ОПЫТ ЭКСПЛУАТАЦИИ ТРАНСЛЯТОРА С СОКРАЩЕННОГО ЯЗЫКА АЛГОЛ-60

Гитман И.Р., Устрайх М.А.

На конференции пользующихся машинами типа "Урал-2" (Одесса 1965 г.) было сообщено о создании транслятора с сокращенного языка АЛГОЛ-60.

В настоящем докладе излагается опыт эксплуатации транслятора.

Рассматриваются следующие вопросы.

1. Характеристики транслятора и программ, составленных транслятором.
2. Организация работы с транслятором.
3. Характеристики задач, отладка, скорость прохождения задач до получения решения.
4. Моделирование процедур и использование ССП-22 и библиотеки стандартных программ.
5. Документация транслятора и передача транслятора другим организациям.

ТРАНСЛЯТОР С АЛГОЛ-60 (*Subset*) ДЛЯ „УРАЛ-2-4“

Горинovich Л.Н.

Транслятор предназначен для машин типа Урал-2-4 и представляет собой программирующую программу, осуществляющую перевод текста задачи на языке АЛГОЛ-60 в программу в кодах машины.

Входным языком, на котором составляются задачи, подлежащие программированию, является сокращенный АЛГОЛ-60 (*Subset*) с некоторыми расширениями: именуемое выражение и переключательный список даны без ограничений.

Имеются стандартные процедуры ввода и вывода величин во внешнюю память.

Транслятор предусматривает возможность использования при решении задач библиотеки стандартных программ в системе автоматизации для машин Урал-2-4.

Транслятор имеет блочную структуру. Все блоки транслятора, кроме вспомогательных блоков, находящихся в НФ, записаны на магнитный барабан. В процессе трансляции необходимые блоки по одному вызываются в НФ программой НС (интерпретирующей системой) и по окончании работы записываются на МБ со всей необходимой информацией.

Текст задачи, записанной на АЛГОЛ-е, набирается на теле-

тайпе ТА-50 на бумажную ленту в 2-м международном коде и вводится в машину.

Программа, полученная при транслировании блока по I300 команд записывается на МБ, и по окончании транслирования собирается в НФ при помощи ССП-22 или используется при помощи ИС.

Предусматривается программа - администратор, полностью автоматизирующая работу на машине при трансляции.

СТАНДАРТНАЯ ПРОГРАММА "ЛОГИЧЕСКАЯ СХЕМА" (Л.С.)

Иоффе П.Д.

Программа Л.С. предназначена для обработки любой блок-схемы. Программа Л.С. включает в себя программу отладки. Для работы программы Л.С. составляется справочный массив, в котором для каждого блока записаны номера блоков, на которые имеются выходы. Все блоки блок-схемы предварительно перенумеровываются в любом порядке.

Программа Л.С. производит на основе справочного массива логическую "сшивку" всех блоков.

Программа Л.С. предусматривает возможность включения в качестве подпрограммы логически законченных блок-схем.

Опробование программы показало, что скорость программирования логических блок-схем увеличивается в 1,2 - 1,5 раз, а время отладки уменьшается в 2-3 раза.

Программа в различных модификациях занимает от 40 до 112 восьмеричных ячеек.

ОПЫТ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ АВТОМАТИЗАЦИИ ПРОГРАММИРОВАНИЯ, ОСНОВАННОЙ НА "ССП-42"

Кудряшов А.Н.

В сообщении рассматриваются следующие вопросы:

1) Удобство и универсальность данной системы автоматизации.

2) Необходимость расширения имеющейся библиотеки стандартных программ и улучшение этих программ. Последнее должно коснуться программ по линейной алгебре для матриц, близких к особым.

3) Сейчас накоплен некоторый опыт работы с библиотекой. Необходима унификация обращения к однотипным программам.

4) Некоторые обслуживающие и стандартные программы, созданные в нашей организации (в частности, программы печати таблиц).

5) Возможность создания на основе данной системы автоматизации трансляторов с логических языков.

ИМИТАТОР АЦВМ "УРАЛ-4" НА АЦВМ "МИНСК-1"

Максимова Н.Н., Орлова Н.В., Рохлин Ф.З. Памгунова В.Н.

I. Задача программного моделирования автоматических цифровых вычислительных машин (АЦВМ) имеет определенное принципиальное и прикладное значение. Она представляет интерес, например, в таких вопросах, как:

а) моделирование проектируемых машин с целью экспериментальной оценки принятых проектировщиками решений; при этом может оцениваться как система команд, так и структурная схема всей машины или ее отдельных блоков;

б) отладка на данной машине программ, составленных в коде другой машины (составление тестов для проектируемых машин; участие нескольких организаций, располагающих различными машинами, в составлении одной программы; обучение программированию на машинах, отсутствующих в данной организации, и т.п.);

в) рациональное распределение функций между машинами различной производительности, а именно: использование малых АЦВМ преимущественно для отладки программ, а больших АЦВМ — для решения задач по отлаженным программам.

2. На практике наиболее часто встречается случай моделирования высокопроизводительных АЦВМ на малых АЦВМ. Хотя большинство трудностей программного моделирования АЦВМ являются

общими, однако названный случай имеет свои особенности, обусловленные, в основном, различием в объемах запоминающих устройств и, прежде всего, различием в объемах оперативных устройств.

3. Моделирующую программу (имитатор) следует строить по типу интерпретирующих программ. Имитатор рассматривает по очереди команды, образующие программу моделируемой машины (псевдопрограмму), и выполняет их так, как они выполнялись бы этой машиной. Для этого выделяются специальные ячейки, имитирующие сумматор и основные регистры устройства управления (регистр команд, счетчик команд, триггер сигнала ω , триггер сигнала переполнения φ , счетчик циклов и индексный регистр, если последние имеются в моделируемой машине, и т.п.).

Оперативная память моделируемой машины имитируется в машине, на которой производится моделирование, следующим образом *). Часть внешней памяти машины В отводится для имитации оперативной памяти машины А. С другой стороны, в оперативной памяти машины В отводится специальное место для высова нужных в данный момент частей оперативной памяти машины А. Это место разделяется на так называемые карманы.

Перед выполнением очередной команды псевдопрограммы проверяется, содержится ли эта команда в одном из карманов или нет. В случае положительного ответа эта псевдокоманда вызывается в ячейку, имитирующую регистр команд. В противном случае из внешней памяти в один из карманов вызывается массив

*) Для краткости первую машину будем называть машиной А, а вторую — машиной В.

ячеек, содержащий эту команду. При этом, если карман, в который должен быть произведен вызов, оказывается занятым, его содержимое предварительно переписывается во внешнюю память на место, с которого в последний раз было произведено заполнение данного кармана. Таким образом, необходима меняющаяся в процессе выполнения псевдопрограммы привязка карманов к ячейкам внешней памяти машины В, имитирующим ячейки оперативной памяти машины А. Для этого каждому карману сопоставляются две ячейки, в которые при каждом заполнении данного кармана записываются адреса оперативной памяти машины А, указывающие границы вызванного в карман массива ячеек.

Аналогично, при выполнении команды псевдопрограммы в первую очередь проверяется, находятся ли в карманах ячейки, адреса которых указаны в данной команде. При положительном ответе производится выполнение команды. В противном случае перед выполнением команды производится заполнение одного из карманов (быть может, нескольких карманов), с предварительным переписыванием, если это требуется, их содержимого в соответствующие места внешней памяти. Для определения того, в какой из карманов (в какие карманы) следует производить вызов (если таковой требуется), предусмотрена специальная система приоритета, делающая на основе анализа предистории работы псевдопрограммы указания о том, содержимое каких карманов скорее всего не понадобится в ближайшее время.

4. Кроме имитации сумматора и регистров устройства управления необходима имитация панели сигнализации и основных кнопок и тумблеров пульта управления. Последние имитируются ячейками оперативной памяти, а имитация панели сигнализации

состоит в том, что после останова (в случае выполнения команды "останов" или в случае "останова по адресу"), печатается содержимое ячеек, имитирующих те регистры устройства управления, которые выведены на панель сигнализации машины А.

5. То обстоятельство, что имитатор выполнен по типу интерпретирующих программ, позволяет достаточно просто вводить различные отладочные режимы работы имитатора, которые особенно полезны в случае, когда имитатор используется для отладки псевдопрограмм.

6. На основе изложенных принципов для АЦВМ "Минск-1" оставлена программа - имитатор, моделирующая АЦВМ "Урал-4". Имеется 4 отладочных режима. Структура имитатора позволяет вводить по мере надобности и другие отладочные режимы.

Двухлетний опыт работы с имитатором подтвердил пригодность его для целей отладки программ.

ИСПОЛЬЗОВАНИЕ ОБСЛУЖИВАЮЩИХ ПРОГРАММ

Махмуд-Заде Р.А.

1. Данной системой удобно пользоваться при содержательном программировании.

2. Все программы по своему назначению делятся на три группы.

а) Обслуживающие программы выполняются с пульта. Все эти программы оканчиваются выходом на "Стоп".

б) Обслуживающие программы, с обращением из основной программы.

в) Рабочие программы, служащие подпрограммами.

3. Все библиотеки по способу вызова делятся на три библиотеки В-0, В-1, В-2.

В-0 - вызывается всегда. Она необходима при решении почти любых задач. К ней относятся следующие программы: вызов КП; перекодировка; вызов программ ввода; перевод 10-2; программа "Туда", "Обратно", (конец работы, начало работы).

Программы "Инфо" (для переноса информации) и т.д.

Информация для программ, работающих с пульта, задается в сумматоре или на ключах. В-0 занимает 60 неполных ячеек, расположенных в конце НР.

К В-1 относятся следующие программы: КП, программа вво-

да и некоторые другие.

Программы Б-1 вызываются и обрабатываются при помощи Б-0. КП может быть вызвана в любое место НД.

К библиотеке Б-2 относятся все рабочие и обслуживающие программы, вызываемые при помощи КП в любое место НД. КП занимает 400 ячеек.

Рассматриваемая система позволит производить компиляцию программы из стандартных массивов.

Работа, необходимая для ввода программы, сведена почти к минимуму.

НЕКОТОРЫЕ СЕРВИСНЫЕ ПРОГРАММЫ ДЛЯ МАШИНЫ "УРАЛ-3"

Мусаев А.М.

Под сервисными (обслуживающими) программами понимаются программы, предназначенные для обеспечения удобной работы программиста за пультом.

1. Программа "Поправка с пульта"

Как показывает практика работы, непосредственное исправление обнаруженной ошибки приводит по различным причинам к новым ошибкам (поправка вводится не в ту ячейку, либо вводится не то машинное слово, которое требуется).

Для упрощения и стандартизации внесения исправлений вручную с пульта, предлагается данная программа.

Программа предназначена для проведения исправлений в оперативном накопителе.

Программа работает следующим образом: прежнее содержимое a_{uc} запоминается в информационной ячейке *Инф3* и в нее записывается содержимое ячейки *Инф2*, т.е. поправка, после чего программа выходит на "Стоп", и на панели сигнализации в сумматоре в разрядах с 7 по 19 будет находиться номер ячейки a_{uc} а в разрядах с 20 по 39 сама поправка. Если поправка внесена неправильно, то повторным нажатием "пуск"

автоматически восстанавливается прежнее содержимое *а_м*.

Второй вариант программы

После ввода всякого массива в оперативную память и во время решения задачи, и во время отладки программы необходимо все содержимое оперативной памяти (кроме нескольких общих рабочих ячеек) переписать на магнитный барабан для исключения необходимости повторных вводов при возникших сбоях и по другим причинам.

Лучший способ проведения исправления ошибки

- а) При обнаружении ошибки в массиве вызвать массив с магнитного барабана с помощью программы "Обратно".
- б) Исправить ошибку с помощью программы "Поправка с пульта".
- в) После исправления ошибки вновь переписать массив на магнитный барабан с помощью программы "Туда".
- г) Продолжать отладку или решение задачи.

Нужно отметить, что введение такой системы работы приведет практически к почти 10-кратному выигрышу во времени отладки.

Во втором варианте объединены программы "Туда", "Обратно" в программе "Поправка с пульта".

2. Выдача информации на печатающие устройства

Для выдачи полученных результатов при решении задачи в удобной для программистов форме предлагаются некоторые программы печати.

Программа "Печать массива чисел с пульта на цифровом печатающем устройстве"

Программа предназначена для печати содержимого группы ячеек оперативного накопителя в десятичной системе счисления по полным ячейкам от ячейки A_n по ячейке A_k , заданной программистом.

Программой производится печать массива по группам по n чисел, где n тоже задается программистом. Перед каждой группой печатается адрес первой ячейки группы и пропускается интервал.

Программа "Печать массива чисел на алфавитно-цифровом печатающем устройстве."

Программа предназначена для печати содержимого группы ячеек оперативного накопителя по полным ячейкам.

Она работает в двух режимах в зависимости от положения ключа K , используемого в программе. При включении ключа K на пульте печать происходит с переводом массива из двоичной системы счисления в десятичную, при выключении — без перевода, т.е. печатается ранее переведенный массив.

Программой производится печать массива по столбикам в

пять столбиков по всей ширине бумаги, заправляемой в АЦПУ. В каждом столбике печатаются n строк (где n задается программистом) каждая из которых имеет вид:

$$\alpha_1 \alpha_2 \alpha_3 \alpha_4 \text{Пр}^k \beta_1 \beta_2 \beta_3 \beta_4 \beta_5 \beta_6 \beta_7 \beta_8 \beta_9 \beta_{10} n_1 n_2$$

где $(\alpha_1 \alpha_2 \alpha_3 \alpha_4)_8$ - адрес ячейки, в которой содержится печатаемое десятичное число, Пр^k - пробел, k - количество пропускаемых пробелов, β_2^k - знак порядка, $(n_1 n_2)_{10}$ - порядок числа, β_1^k - знак мантиисы, $(\beta_1 \beta_2 \beta_3 \beta_4 \beta_5 \beta_6 \beta_7 \beta_8 \beta_9 \beta_{10})_{10}$ - мантииса числа. Между строками столбиков оставляется пять пробелов.

Программа может включаться в работу как с пульта, так и передачей управления из основной программы.

Программа "Печать программы на АЦПУ"

Она производит печать массива в коде команд по столбикам в шесть столбиков по всей ширине бумаги, заправляемой в АЦПУ. В каждом столбике печатаются n строк (n - задается программистом), каждая из которых имеет вид:

$$\alpha_1 \alpha_2 \alpha_3 \alpha_4 \text{Пр}^k \beta_1 n_1 n_2 \text{Пр}^j \beta_2 \beta_3 \beta_4 \text{Пр}^i \beta_5$$

СТАНДАРТНАЯ АППРОКСИМИРУЮЩАЯ ПРОГРАММА (СТАПП) ПО МЕТОДУ НАИМЕНЬШИХ КВАДРАТОВ (МНК) ПАРАБОЛОЙ n -ГО ПОРЯДКА

Новоселов В.С.

Постановка задачи:

а) Даны N значений y_1, \dots, y_N некоторой функции $y=f(x)$ при значениях x соответственно x_1, \dots, x_N . Найти значения параметров a_0, a_1, \dots, a_n , минимизирующих квадратичную форму
$$S = \sum_{i=1}^N \left(y_i - \sum_{k=0}^n a_k x_i^k \right)^2 p_i \quad (p_i - \text{заданные неотрицательные числа}).$$

б) Считая y_i значениями нормальных случайных величин с математическими ожиданиями $My_i = P(x_i), P(x) = \sum_{k=0}^n a_k x^k$, и дисперсиями σ_i^2 , найти оценки $\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_n$ величин a_0, a_1, \dots, a_n по МНК, т.е. минимизирующие то же самое выражение S при $p_i = \frac{\sigma_i^2}{\sigma_i^2}$ (σ_i^2 - константа, выбираемая из соображений числового удобства).

Для этих оценок определить доверительные интервалы, точечные оценки \tilde{y}_i и ковариационную матрицу $\hat{B}_{\tilde{A}}$ для оценок $\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_n$.

Как видно, решение задачи а) предоставляет собой часть решения задачи б). Для решения задачи б) составлена программа СТАПП.

Общее решение.

Как известно, задача нахождения $\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_n$ сводится к

решению системы нормальных уравнений (СНУ, система линейных алгебраических уравнений $(n+1)$ -го порядка), имеющих для параболической аппроксимации вид $XA=B$, где

$$X = \begin{pmatrix} [px^0, x^0] & [px, x] & \dots & [px^n, x^n] \\ [px, x] & [px^2, x^2] & \dots & [px^{n+1}, x^{n+1}] \\ \dots & \dots & \dots & \dots \\ [px^n, x^n] & [px^{n+1}, x^{n+1}] & \dots & [px^{2n}, x^{2n}] \end{pmatrix}, B = \begin{pmatrix} [px, y] \\ [px, y] \\ \dots \\ [px, y] \end{pmatrix}, A = \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_n \end{pmatrix}$$

A — неизвестные, и обозначено $[px^k, z^c] = \sum_{i=1}^n p_i x_i^k z_i^c$, если $z = (z_1, \dots, z_n)$ — произвольный вектор. Оценки $\tilde{a}_0, \tilde{a}_1, \dots, \tilde{a}_n$ представляют собой решение этой системы. Доверительные интервалы для \tilde{a}_k равны $\tilde{a}_k \pm \gamma \sigma \tilde{a}_k$, где γ выбирается из таблицы распределения Стюдента для 95% уровня значимости и $N-n-1$ степеней свободы, $\sigma_{\tilde{a}_k}^2(x^{-1})_{kk} \sigma^2 = (B_{\tilde{A}})_{kk}$, запись $(c)_{kk}$ означает k -ый диагональный элемент матрицы C .

Описание СТАПШ.

Программа рассчитана на $n \leq 10$. Время работы для $n=10$ примерно 12 мин. Для $n=5$ — около 3 мин. СТАПШ состоит из двух частей: стандартной и нестандартной.

Стандартная часть СТАПШ (в дальнейшем говорится СТАПШ, подразумевая только ее стандартную часть) занимает вместе с рабочими ячейками 1300₈ ячеек (неполных). Она включает все основные расчеты по приведенным выше формулам.

Нестандартная часть составляется отдельно для каждого конкретного класса задач.

Работа СТАПШ начинается после того, как в оперативной памяти размещены в нормализованном двоичном виде с плаваю-

шей занятой массивы x, y, p . Для возможности формирований в нестандартной части предусмотрено обращение из нестандартной части к СТАП с возвратом.

СТАП рассчитана на применение ССП-22 и использует ряд библиотечных стандартных программ (перевод чисел, некоторые элементарные функции и др.) с фиксированными номерами массивов.

Некоторые особенности СТАП.

1. Поскольку в матрице СЛУ участвуют суммы k -ых степеней x -ов, то при не очень "хороших" данных могут исчезнуть при счете малые x -ы (например, x^{2n} при $n=10: x^{20}$, т.е. при $x \sim 10^{-1}$, x^{20} — машинный 0), что эквивалентно потере экспериментальных точек, или наоборот, может произойти переполнение ($n=10$, x^{20} , переполнение при $x \sim 10^1$). Поэтому СТАП осуществляет т.н. "нормализацию" x -ов и y -ов, обеспечивающую максимальную "сохранность" материала, с одной стороны, при условии отсутствия переполнения, с другой.

Соответственно, после счета всех оценок (\tilde{a}_k , доверительных интервалов, $B_{\tilde{A}}$) следует произвести обратную операцию, (т.н. "денормализация") чтобы получить значения нужных величин в прежнем масштабе. Эта операция также выполняется в СТАП.

2. В целях достижения компромисса между объемом и сложностью программы, объемом занятой памяти и временем работы, при счете матрицы СЛУ используется массив N рабочих ячеек (полных) для величины x_k^* (k — фиксируется).

3. Используется программа решения системы линейных уравнений методом Гаусса с выбором главного элемента и не используется программа обращения матриц, даже для счета обратной

матрицы ($B_{\bar{A}} = x^{-1} \sigma^2$). Это сделано из тех соображений, что метод Гаусса наименее чувствителен к эффектам плохой обусловленности матрицы системы.

Довольно значительная длина СТАП (1100 ячеек программы) обусловлена, в частности, влиянием вычислений, связанных с "нормализацией" и "денормализацией", удлиняющих программу примерно в полтора раза.

Небольшое удлинение вызывает также предпочтение метода Гаусса вместо обращения матрицы.

Заключение.

СТАП применялась для расчетов констант комплексообразования некоторых соединений (в радиохимии) и некоторых других физико-химических констант. Опыт работы показал существенное влияние весов на результат.

БЕССКОБОЧНАЯ ФОРМА ЯЗЫКА АЛГОЛ-60 И АЛГОРИТМ ПЕРЕВОДА

Орлов Н.К., Джорбенадзе Э.Г.

При использовании в трансляторах бесскобочной формы записи, обычно ограничиваются переводом в эту форму арифметических выражений. При этом оказывается, что арифметические выражения переводятся в машинную форму при помощи алгоритма стекового типа, а для перевода всего остального используются алгоритмы других типов. Но если перевести в бесскобочную форму весь входной язык, то транслятор может быть построен в виде единого алгоритма стекового типа.

В настоящей работе предлагается подобная бесскобочная форма для всех операторов АЛГОЛа, учитывающая возможность рекурсивного вхождения операторов друг в друга и алгоритм перевода в эту форму. Синтаксис АЛГОЛа при этом изменяется; каждый ограничитель АЛГОЛа после "перетасовки" рассматривается как некоторый оператор. Для того чтобы избавиться от необходимости анализировать запись вперед и назад, структура языка построена так, что анализ вперед не требуется, а операторы /ограничители/ модифицируются, приобретая дополнительные признаки в зависимости от того, что им предшествовало. Модифицированные операторы рассматриваются как новые операторы. Действие каждого оператора

однозначно.

Перевод в бесскобочную форму осуществляется при помощи алгоритма стекового типа с использованием системы приоритетов, присваиваемых ограничителям АЛГОЛа. Кроме этого в этом же стеке осуществляется модификация операторов /ограничителей/.

Алгоритм реализован на УЦМ УРАЛ-2 и составляет часть транслятора с языка АЛГОЛ-60.

СТЕКОВЫЙ АЛГОРИТМ ПЕРЕВОДА С БЕССКОБОЧНОЙ ФОРМЫ ЯЗЫКА АЛГОЛ-60 НА ЯЗЫК УЦВМ УРАЛ-2

Орлов Ю.К., Джорбенадзе Э.Г.

Алгоритм переводит с бесскобочной формы языка АЛГОЛ-60 в последовательность команд УЦВМ УРАЛ-2. Перевод осуществляется в линейной последовательности без анализа исходной записи вперед и назад, без каких-либо изменений уже записанных частей программы, за исключением заполнения адресной части в командах передачи управления по адресам, которые в момент формирования этих команд не были известны.

Все идентификаторы и константы в бесскобочной форме АЛГОЛа рассматриваются как операнды, а все ограничители (в т.ч. и модифицированные) — как операторы. Идентификаторы поступают в стек с двумя выходными регистрами, а ограничители управляют состоянием регистров стека, формированием команд и составлением таблицы идентификаторов.

Действие каждого ограничителя-оператора однозначно и не зависит ни от предшествующих, ни от последующих ограничителей.

Составляя таблицу идентификаторов, алгоритм распределяет память, присваивая каждому идентификатору действительный адрес и фиксируя некоторый набор сведений

о нем, определяемый управляющим этим процессом ограничителем (обычно описателем или спецификатором). Распределение памяти начинается с высших адресов НФ. Программа при этом начинается с низших адресов, благодаря чему достигается максимальное использование памяти.

Каждый ограничитель генерирует строго фиксированную группу команд, заполняя адресные части в соответствии с содержимым регистра стекла и данными таблицы идентификаторов. Ограничитель, обозначающий конец блока, вызывает стирание части таблицы идентификаторов, граница этого стирания определяется адресом, появляющимся к этому моменту в стеке.

ТРАНСЛЯЦИИ ТЕЛ ПРОЦЕДУР И ОПЕРАТОРОВ ПРОЦЕДУР ПРИ ЗАПИСИ АЛГОРИТМА В БЕССКОБОЧНОЙ ФОРМЕ АЛГОЛА

Орлов Ю.К., Леонидзе Ю.Т., Джорбенадзе Э.Г.

Транслятор распределяет память, отводя по ячейке на каждый формальный параметр процедуры, но программирование тела процедуры ведется с учетом того, что в эти ячейки записывается либо значение параметра, либо информация о нем, если он должен быть задан наименованием. Это вызывает изменение стандартных групп команд, генерируемых некоторыми ограничителями; изменение это обуславливается содержанием таблицы идентификаторов.

Обращение к процедуре осуществляется при помощи программы реализации процедур, которая вводится в память одновременно с переведенной транслятором программой. Программа реализации процедуры пользуется информацией о процедуре, расположенной перед началом тела процедуры.

ПЕРЕВОД ЗАКОДИРОВАННОЙ ЗАПИСИ НА ЯЗЫКЕ АЛГОЛ-60
НА ВНУТРЕННИЙ ЯЗЫК ТРАНСЛЯТОРА И ВЫДЕЛЕНИЕ ОСНОВНЫХ
ПОНЯТИЙ АЛГОЛА

Орлов Ю.К., Микшашивили Г.Д.

Основной особенностью предлагаемого алгоритма перевода является отсутствие различения между строчными и заглавными буквами в АЛГОЛе /при этом, естественно, идентификаторы не должны совпадать побуквенно с буквенными ограничителями АЛГОЛа/. Считывая по одному основные символы АЛГОЛа/ в их число включен и символ "типографский пробел"/, алгоритм выделяет следующие элементарные понятия: ограничители, идентификаторы и константы. На кодировку основного символа приходится 8 двоичных разрядов, на кодировку элементарного понятия - 10 разрядов. При этом встречающиеся константы переводятся в машинную форму. Коды идентификаторов и констант подбираются так, что впоследствии могут служить для формирования действительных адресов соответствующих ячеек.

СТАНДАРТНЫЕ ОТЛАЖИВАЮЩИЕ ПРОГРАММЫ

Павловская Р.Н., Гладкова Т.В.

Приводятся стандартные программы отладки и прокрутки, применение которых значительно упрощает и сокращает процесс отладки.

Сообщается об опыте использования этих программ.

АВТОМАТИЧЕСКИЙ ПЕРЕВОД ПРОГРАММ С ЯЗЫКА ЭВМ "УРАЛ-1" НА ЯЗЫК "УРАЛ-4"

Перская Л.Ф., Вайсбурд Р.А.

В Вычислительном центре Уралмашзавода имеются две машины Урал-1 и Урал-4. Урал-4 работает с января 1966 г. Для наиболее полной и рациональной загрузки этих машин была составлена программа, которая автоматически переводит достаточно широкий класс программ с фиксированной запятой из кодов машины Урал-1 в коды машины Урал-4. Задачи, требующие трудоемкой отладки и последующего длительного или периодического решения предполагается доводить вначале на машине Урал-1 и затем автоматически переводить на Урал-4.

Программа состоит из 2 частей. В первой части осуществляется преобразование перерабатываемой программы таким образом, что все константы располагаются за программой, а все выносы вносятся в программу.

Благодаря этому программы, которые после отладки имеют много выносов и нерациональное использование памяти приводятся к наиболее компактному и логически стройному виду. При этом в массив констант дополнительно включаются константы, появившиеся в результате замены команд 20K на 02 < K >.

Во второй части программы осуществляется замена команд 17a, 05a и др. и преобразование команд 21k и 22k.

Программа отлаживалась на машине Урал-2 и при незначительной корректировке может быть использована для перевода программ из кодов машины Урал-1 в коды машин Урал-2 и Урал-3.

ЦИКЛ ПРОГРАММ ДЛЯ УПОРЯДОЧЕНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ЧИСЕЛ

Перфильева А.И.

Реализуется алгоритм № 175 из журнала "САСМ" 1963 г. и алгоритм §§ 44, 45 и § 47 из книги Г. Боттенбруха "Структура Алгол-60 и его использование". Первая программа производит расстановку массива в определенном /возрастающем или убывающем/ порядке методом перестановки двух соседних чисел. Общее количество операций порядка n^2 . Программа эффективна для сравнительно небольшого /25-50/ массива чисел. В качестве дополнительной памяти используется лишь одна рабочая ячейка. Объем программы - 66 команд.

Следующая группа программ предназначена для упорядочения больших массивов чисел методом, предложенным фон Нейманом и Гольдштайном, заключающимся в разбиении всего массива на последовательности, длина которых возрастает по геометрической прогрессии. Общее количество операций порядка $n \log_2 n$. В качестве дополнительной памяти используется n рабочих ячеек. Объем программ - 200 команд.

АВТОМАТИЗАЦИЯ ПРОГРАММИРОВАНИЯ ДЛЯ ЭЕМ "УРАЛ-4"

Сизов Б.В., Самойлова Г.А.

В сообщении рассматривается система автоматического программирования процессов механической обработки плоских деталей на станках с программным управлением.

1) Программирование процессов механической обработки деталей на станках с программным управлением. Краткая технологическая схема процесса программирования механообработки.

2) Вопросы автоматизации процессов программирования механообработки.

3) Система автоматического программирования процесса механообработки контуров плоских деталей, составленных отрезками прямых и дугами окружностей:

- а) язык системы и технологические данные,
- б) блок-схема системы,
- в) работа системы.

4) Заключение. Усовершенствование и развитие систем автоматического программирования.

ПРОГРАММЫ УПОРЯДОЧЕНИЯ МАССИВОВ, ПРЕВЫШАЮЩИХ ПО ВЕЛИЧИНЕ ОПЕРАТИВНУЮ ПАМЯТЬ

Устинов Н.П.

Нетрудно представить множество объектов, которые обладают либо все одинаковым набором свойств, либо частью свойств из всех возможных.

Например: 1. Каждый материал в производстве, идущий на изготовление какого-либо изделия, характеризуется номенклатурным номером, единицей измерения, ценой, плановым и фактическим запасом.

2. При начислении зарплаты каждый работающий характеризуется в расчетном листке только теми видами оплаты и удержания из всех возможных, которые ему положены в расчетном месяце. Сведения об одном объекте множества называют сообщением / фразой или строкой. Сообщение состоит из характеристик отдельных свойств описываемого объекта - реквизитов / показателей /. Различают реквизиты количественные и признаковые. В дальнейшем признаковый реквизит мы будем называть, для удобства, коротко - признаком. При размещении сообщения в единицах емкости носителя принимается определенный макет (схема) расположения реквизитов. При этом для размещения

сообщений с переменным числом реквизитов необходимо ввести понятие переменный макет, т.е. совокупность возможных макетов с допустимыми сочетаниями реквизитов.

Из вышесказанного /в зависимости от постоянного или переменного числа свойств у объектов множества/можно различать массивы сообщений соответственно с постоянным или переменным макетом.

В ВЦ Красноярского алюминиевого завода для ЭВМ "Урал-4" разработаны программы упорядочения хранимых на магнитном барабане массивов, превышающих по величине оперативную память.

Упорядочение производится методом слияния, т.е. массив разделяется на части, которые упорядочиваются в оперативной памяти, а на втором этапе упорядоченные части сливаются до тех пор, пока не получится один упорядоченный массив.

Этап разделения массивов на части и их упорядочения в оперативной памяти выполняется поразрядной сортировкой. Алгоритм упорядочения в оперативной памяти проследим на примере.

Признаки исходного массива	Упорядочение исходного массива по I-му /справа/ разряду признака
I 0 I	0 0 0
0 0 I	0 0 0
0 0 0	0 I 0
I I I	I 0 I
0 0 0	0 0 I

0	0	I
0	I	0
0	0	I
0	I	I

Упорядочение полученного массива по второму разряду признака

0	0	0
0	0	0
I	0	I
0	0	I
0	0	I
0	0	I
0	I	0
I	I	I
0	I	I

I	I	I
0	0	I
0	0	I
0	I	I

Упорядочение вновь полученного массива по третьему разряду признака

0	0	0
0	0	0
0	0	I
0	0	I
0	0	I
0	I	0
0	I	I
I	0	I
I	I	I

На ЭММ этот алгоритм реализуется отбором на свободное поле в оперативной памяти сообщений с **нулем** а затем - с единицей в рассматриваемом разряде. Сообщения размещаются в полных ячейках. Начало переменного макета определяется по единице в нулевом разряде полной ячейки. В переменном макете признак, по которому упорядочивается массив, может располагаться с I по 39 разряды первой полной ячейки, а в постоянном макете - с 0 по 39 разряды любой полной ячейки этого макета.

Алгоритм упорядочения выполняется двумя самостоятельными программами, чтобы большую часть оперативной памяти

отвести под упорядочиваемую информацию и тем самым уменьшить время сортировки массива. Одна программа выполняет разделение массива на части и упорядочивает их в оперативной памяти, а вторая программа попарно сливает упорядоченные части в один упорядоченный массив.

В настоящее время составлены две пары программ: одна — для упорядочения массивов с постоянным макетом и вторая — для упорядочения массивов с переменным макетом.

При обращении к программам кусочного упорядочения массивов задаются следующие исходные данные:

1. Константа выделения признака.
2. Количество ячеек в макете /для массивов с постоянным макетом/.
3. Место признака в макете /задается количеством неполных ячеек от начала макета до полной ячейки с признаком/.

Примечание: Место признака в макете задается только для массивов с постоянным макетом. У массивов с переменным макетом признак всегда в первой полной ячейке.

4. Начальный адрес и величина исходного массива на магнитном барабане.
5. Начальный адрес на магнитном барабане, с которого размещаются сведения об упорядоченных частях массива.

При обращении к программам слияния упорядоченных частей массива дополнительно задается начальный адрес свободного поля на магнитном барабане и количество сливаемых частей

массива.

Время упорядочения прямо пропорционально количеству двоичных разрядов в признаке. В среднем в минуту упорядочивается от 1000 до 3000 полных ячеек.

СТАНДАРТНАЯ ПРОГРАММА СОРТИРОВКИ ИНФОРМАЦИИ НА НМЛ

Ханамиров С.С.

При решении задач по учету и оперативному планированию на ЭЦМ "Урал-4" часто возникает необходимость расположить имеющуюся информацию в порядке возрастания какого-либо одного или нескольких признаков, т.е. рассортировать информацию.

В Вычислительном центре ХТЗ им. С.М. Кирова разработан алгоритм и составлена стандартная программа сортировки информации на магнитной ленте. Ее назначение и основные характеристики следующие:

Пусть на НМЛ записана информация, состоящая либо из отдельных чисел, либо из групп по несколько чисел, или, иначе, пусть на НМЛ записана матрица по столбцам, состоящая из одной или нескольких строк. Требуется так перерасположить столбцы матрицы, чтобы в заданной, например в i -ой, строке установилось возрастание по некоторому признаку. Одновременно, в тех подмассивах, в которых после упорядочения соберутся столбцы с одним и тем же сортируемым признаком, нужно установить возрастание по другому признаку (он может быть в другой строке матрицы). То есть требуется произвести двуступенчатую сортировку. Размеры массива ограничиваются ёмкостью НМЛ, и всю информацию обозреть за один раз не представляется возможным. Использование операции выборки в этом случае мало-

эффективно.

Указанная программа исполнена в виде стандартной. С ее помощью можно осуществить одно-, двух-, трех-, четырехступенчатую сортировку. Обращение к программе занимает несколько ячеек (их количество зависит от количества ступеней сортировки). В этом обращении указывается, где находится на ИМЛ информация, подлежащая сортировке, что можно занимать в ИФ для работы программы и как нужно сортировать (количество ступеней сортировки, с каким шагом и по каким признакам).

Алгоритм, используемый в программе, может быть реализован в аналогичной программе сортировки информации на ИМБ.

ПРИМЕНЕНИЕ АЛГОРИТМИЧЕСКОГО ЯЗЫКА АЛГОЛ-60 ПРИ ПРОГРАММИРОВАНИИ ТЕХНИЧЕСКИХ ЗАДАЧ

Шарова Л.Д.

В процессе программирования задача, записанная на математическом языке, переводится на язык машины. Получающиеся при таком переводе программы часто оказываются весьма длинными и не имеют той наглядности, которую имела первоначальная запись задачи. Это влечет за собой увеличение вероятности ошибок, которые выявляются после длительной отладки, но нередко остаются невыявленными и после отладки. В результате полезность ЭЦВМ значительно снижается из-за трудностей и больших затрат на программирование и отладку задач.

Возникла задача автоматизации программирования. С этой целью был создан ряд промежуточных языков для записи алгоритмов. Одним из таких языков является АЛГОЛ-60. Он имеет ряд преимуществ по сравнению с остальными. Это международный язык, который служит обмену алгоритмами между людьми. В программе, записанной на АЛГОЛе, четко выражен процесс вычислений в форме, удобной для программирования. Созданы трансляторы (программирующие программы), которые анализируют описания, сделанные на алгоритмическом языке, и перерабатывают их в программу, состоящую из машинных команд.

Многие организации занимаются созданием трансляторов с

АЛГОЛа-60 на язык различных машин.

В нашем институте имеется транслятор для машины "Урал-4", с помощью которого были получены результаты нескольких программ, записанных на АЛГОЛе.

Так были получены результаты работы "Технико-экономическое сравнение вариантов турбогенераторов мощностью 800 Мвт с $\cos \varphi_n = 0,85$ и $\cos \varphi_n = 0,9$ ".

Сравнивались варианты, отличающиеся:

1. по напряжению - 330 и 550 кв;
2. по длине линии - 100, 200, 300, 400, 500 км.

Критерием сравнения являлись расчетные затраты.

АВТОМАТИЗАЦИЯ ОТЛАДКИ ПРОГРАММ НА ЭЦВМ "УРАЛ-2" МЕТОДОМ ИНТЕРПРЕТАЦИИ

Щербина Ю.В., Пенькова Н.Е., Гонтар М.В., Колбина В.И.

1. Существуют различные подходы к автоматизации отладки сложных программ на ЭЦВМ. Для практической работы наиболее удобен метод интерпретации (I), когда в памяти ЭЦВМ программным путем интерпретируется "вторая" ЭЦВМ. Тогда исполнение отлаживаемой программы поручается интерпретированной ЭЦВМ, а реальная ЭЦВМ "следит" за работой интерпретированной ЭЦВМ и печатает на бумажную ленту информацию, которую мог бы получить программист путем просмотра работы отлаживаемой программы в однократном режиме.

2. Разработка таких отладочных программ для ЭЦВМ "Урал-2" затрудняется некоторыми конструктивными особенностями машины (невозможно записать в память содержимое регистра команд и др.). Однако ряд удачно найденных программистских решений позволил разработать алгоритм полной интерпретации ЭЦВМ "Урал-2" в памяти ЭЦВМ "Урал-2" и составить соответствующую отладочную программу (условное обозначение СПО-4).

3. Практическое использование СПО-4 отличается предельной простотой. Программист имеет возможность довести решение до любой команды отлаживаемой программы и после остановки по адресу включить в работу СПО-4. Решение будет продолжаться с

печатью результатов работы СПО-4 до новой остановки по адресу, где можно отключить СПО-4 и продолжить решение без ее участия.

4. СПО-4 может интерпретировать одноктактный режим работы ЭЦВМ "Урал-2". При этом состояние всех регистров ЭЦВМ после остановки (кроме "номера операции") получается таким же, как и при работе без СПО-4.

5. Числа с плавающей запятой СПО-4 печатает в десятичной системе счисления.

6. Имеется блокировка от затирания СПО-4 командами или числами отлаживаемой программы. Ввод СПО-4 может производиться в любое место оперативной памяти (есть оператор автонастройки, обеспечивающий привязку СПО-4 к истинным адресам). Длина программы - 500_8 неполных ячеек.

7. Программа СПО-4 внедрена в КИМ и ЛКТИ и обеспечивает значительное ускорение отладки программ.

Литература.

Г. Шербина Ю.В., Комлач В.И., Зорина З.С., Методы автоматизации отладки сложных программ работы ЦВМ, Сборник "Вычислительная техника в проектировании и эксплуатации энергосистем", Издательство "Наукова думка", Киев, 1964.

УНИВЕРСАЛЬНАЯ ПЕЧАТАЮЩАЯ ПРОГРАММА

Яегер А.А.

Универсальная печатающая программа (УПП) печатает результаты по форматам, которые для каждого конкретного случая задается программистом.

Поэтому информация для УПП состоит в общем случае из двух частей:

описание выходного массива — определяет, из каких ячеек берутся числа (результаты), которые нужно печатать;

описание формата — определяет, каким образом результаты должны быть расположены на бумажной ленте печатающего устройства.

Для задания информации выработан входный язык, при помощи которого можно описать произвольные форматы.

УПП состоит из следующих частей:

анализирующая подпрограмма,

служебная подпрограмма,

подпрограмма печати.

БЛОК ТРАНСЛЯТОРА ДЛЯ НАПИСАНИЯ ПСЕВДОКОДОВ АРИФМЕТИЧЕСКИХ ВЫРАЖЕНИЙ

В вычислительном центре предприятия разработан транслятор для "Урала-2" с сокращенного языка "Алгол-60".

Транслятор состоит из блоков - синтаксического анализа; распределения памяти; написания псевдокодов арифметических выражений; условных выражений, процедур; блок программирования написанных псевдокодов. Как наиболее удачный предлагается блок написания псевдокодов арифметических выражений. Псевдокод - это промежуточный язык. Для каждой операции пишется один псевдокод, состоящий из номера псевдокода, операции и двух аргументов. Псевдокоды пишутся с конца арифметического выражения. Для одинаковых операций и аргументов, встречающихся несколько раз в арифметическом выражении, псевдокод пишется один раз. То же в случае, если аргументы отличаются знаками. В псевдокоде указывается тип результата */integer, real /*.

СО Д Е Р Ж А Н И Е

Алаев Г.Т. Программа печати карт в изолиниях на ЭВМ "Урал-4"	3
Анисимов В.Я., Лозница В.С., Козлик Л.Г., Тодавчин А.Ю., Буханевич М.Г. Система автоматизации программирования	5
Астахов Ю.И., Перфильева А.И. Опыт использования интерпретирующей программы для ЭВМ "Урал-2"	6
Астахов Ю.И., Сергеева Е.С. Обслуживающие программы для ЭВМ "Урал-2"	7
Афанасьева Г.И., Коростовешская Ю.М. Преобразование цифровой покомпонентной информации в строчную	8
Афанасьева Г.И., Коростовешская Ю.М. Операции с числами, превышающими разрядную сетку "Урал-4", с сохранением точности	9
Афанасьева Г.И., Коростовешская Ю.М. Сортировка учетно-статистической информации	10
Афиногенова Е.П., Альбрехтен Л.М. Отладка программы при ручном способе программирования	11
Барашников . . . Программа "СТОРОЖ" для хранения больших массивов экономической информации на "Урал-4"	13
Барлас Р.А. Программирующая программа СПИР	14
Бигдай В.П. Автодиспетчер для отладки и счета программ небольших размеров	15
Борщев В.Б. Универсальная программа синтаксического анализа для трансляторов с языков, заданных графовыми грамматиками	17

Бушко-Жук М.М., Гонца М.Г., Соловьева И.А., Фесенко В.И. Об усовершенствовании и внедрении транслятора с адресного языка на машину "Урал-2" ("Урал-4")	18
Бушко-Жук М.М., Фесенко В.И. О синтаксическом методе проверки формальной правильности транслируемых алгоритмов	19
Гамецкая С.П., Данильченко А.Ф. Реализация синтаксического метода проверки формальной правильности транслируемых алгоритмов на ЭВМ "Урал-2"	20
Гитман И.Р., Устрайх М.А. Опыт эксплуатации транслятора с сокращенного языка АЛГОЛ-60	21
Гориневич Л.Н. Транслятор с АЛГОЛ-60 (<i>Subset</i>) для "Урал-2-4"	22
Иоффе П.Д. Стандартная программа "ЛОГИЧЕСКАЯ СХЕМА" (Л.С.)	24
Кудряшов А.Н. Опыт использования системы автоматизации программирования, основанной на "ССП-42"	25
Максимова Н.Н., Орлова Н.В., Рохлин Ф.З., Шамгунова В.Н. Имитатор АЦВМ "Урал-4" на АЦВМ "Минск-Г"	26
Махмуд-Заде Р.А. Использование обслуживающих программ	30
Мусаев А.М. Некоторые сервисные программы для машины "Урал-3"	32
Новоселов В.С. Стандартная аппроксимирующая программа (СТАПП) по методу наименьших квадратов (МНК) парабол n -го порядка	36
Орлов Ю.К., Джорбенадзе Э.Г. Бесскбочная форма языка АЛГОЛ-60 и алгоритм перевода	40

Орлов Ю.К., Джорбенадзе Э.Г. Стековый алгоритм перевода с бесскобочной формы языка АЛГОЛ-60 на язык ЭЦВМ "Урал-2"	42
Орлов Ю.К., Леонидзе Ю.Т., Джорбенадзе Э.Г. Трансляция тел процедур и операторов процедур при записи алгоритма в бесскобочной форме АЛГОЛа	44
Орлов Ю.К., Микиашвили Г.Л. Перевод закодированной записи на языке АЛГОЛ-60 на внутренний язык транслятора и выделение основных понятий АЛГОЛа	45
Навловская Р.Н., Гладкова Т.В. Стандартные отлаживающие программы	46
Перская Л.Ф., Вайсбурд Р.А. Автоматический перевод программ с языка ЭВМ "Урал-1" на язык "Урал-4"	47
Перфильева А.И. Цикл программ для упорядочения последовательности чисел	49
Сизов Б.В., Самойлова Г.А. Автоматизация программирования для ЭВМ "Урал-4"	50
Устинов Н.П. Программы упорядочения массивов, превышающих по величине оперативную память . . .	51
Ханамиров С.С. Стандартная программа сортировки информации на НМЛ	56
Шарова Л.Л. Применение алгоритмического языка АЛГОЛ-60 при программировании технических задач	58
Шербина Ю.В., Пенькова Н.Е., Гонтар М.В., Колбина В.И. Автоматизация отладки программ на ЭЦВМ "Урал-2" методом интерпретации	60
Яегер А.А. Универсальная печатающая программа	62
. Блок транслятора для написания псевдокодов арифметических выражений	63

Тартуский государственный университет
ЭССР, г. Тарту, ул.Кликооли, 18

ТЕЗИСЫ ДОКЛАДОВ
У ВСЕСОЮЗНОГО СОВЕЩАНИЯ ПОЛЬЗОВАТЕЛЕЙ
ЭВМ ТИПА "УРАЛ"

Секция II
АВТОМАТИЗАЦИЯ ПРОГРАММИРОВАНИЯ.
ТРАНСЛЯТОРЫ С АЛГОРИТМИЧЕСКИХ ЯЗЫКОВ.
ОБСЛУЖИВАЮЩИЕ ПРОГРАММЫ

На русском языке

Ответственный редактор В.Алласалу
Корректор С.Ирвадин

Репродукция ТГУ 1966. Печ. листов 4,13 (условных 3,76)
Учетн.-издат. листов 1,6. Тираж 800 экз.
Бумага 30х42. 1/4. Сдано в печать 26/VI 1966.
ИЗ-05414. Заказ № 304.
Цена 11 коп.

